

平成30年度  
修士論文

3次元アセンブリモデルを対象としたレイキャ  
スティング法に基づく投影方法とそれを用いた  
検索方法の開発

首都大学東京大学院 システムデザイン研究科  
システムデザイン専攻 情報通信システム学域  
17890516 久米 貴大  
指導教員 片山 薫 准教授



## 目次

<b>1</b>	<b>はじめに</b>	<b>1</b>
<b>2</b>	<b>関連研究</b>	<b>3</b>
<b>3</b>	<b>準備</b>	<b>4</b>
3.1	投影画像計算	4
3.2	特徴量計算	5
3.3	類似度計算	5
<b>4</b>	<b>頂点データからの透過投影画像作成処理</b>	<b>7</b>
4.1	はじめに	7
4.2	比較手法	7
4.2.1	頂点データからボクセルデータへ変換する方法	7
4.2.2	レイキャスティング法	7
4.2.3	スキャンライン法	8
4.3	提案手法	8
4.3.1	画素値の計算	10
4.3.2	内外判定	10
4.4	評価実験	11
4.4.1	使用するモデル	11
4.4.2	処理時間の比較	11
4.4.3	メモリ使用量の比較	12
<b>5</b>	<b>投影作成処理の改善</b>	<b>13</b>
5.1	はじめに	13
5.2	提案手法	13
5.2.1	微小ポリゴンの処理	13
5.2.2	ポリゴン境界の処理	13
5.2.3	同次変換行列の利用	13
5.3	評価実験	14
5.3.1	使用するモデル	14
5.3.2	処理時間の比較	14
<b>6</b>	<b>大規模なモデルの検索</b>	<b>16</b>
6.1	はじめに	16
6.2	大規模なアセンブリモデルの特徴量計算	16
6.3	評価実験	17
6.3.1	使用するモデル	17
6.3.2	処理時間の比較	19
6.3.3	正答率	21
6.3.4	誤差	21

---

6.4 同一形状を持つ大規模なモデルの検索 . . . . .	23
<b>7 結論</b>	<b>24</b>
<b>8 謝辞</b>	<b>25</b>

---



## 1 はじめに

近年，製造業などでは製品の設計やシミュレーションに CAD が普及しており，製品の設計図となり形状を表す CAD モデルデータも数多く作られ続けている．また個人用コンピュータの性能向上や 3D プリンタの普及によって 3 次元形状モデルの活用範囲も身近なものとなっている．蓄積されたモデルデータを効率よく利用するためにはモデル情報の検索が必要である．既存モデルが格納されたデータベースから任意のモデルを取り出すことができれば，新しいモデルを開発するとき既存モデルの情報をを用いて効率よく設計することができる．

こうした背景から 3 次元モデルの検索に関する研究は多く行われているが，そのほとんどはモデルの外見やモデル全体の形状に着目したものである．しかし製品設計で用いられるモデルは複数の部品が組み合わさって構成されるアセンブリモデルであることが多く，外見では判断できない内部構造が存在する場合や微細な部品の配置のみが異なる場合もある．そのためアセンブリモデルの検索においては，従来の検索手法では十分とは言えない．

先行研究として，モデルの形状だけでなく構成部品の材質及び配置を考慮した 3 次元アセンブリモデルの検索手法として，モデルの透過投影画像から特徴量を求める手法がある [1]．微小な部品の配置の違いを強調するためにサブアセンブリの投影画像に対して体積に応じた重みづけを行い，この投影画像をモデルの周囲から多数作成しラドン変換とフーリエ変換を行うことにより，モデルの回転と平行移動に対して頑健な特徴量を得ることができる．しかしこの手法においてはモデルの頂点データからボクセルデータへの変換処理において，投影画像の 1 辺の大きさの 3 乗に比例した処理時間と記憶領域が必要となる．そのため，生成可能な投影画像の解像度に制限があるという問題があった．

そこで本研究では，3 次元アセンブリモデルの透過投影画像を作成処理の計算量削減を目的として，頂点データから投影画像を作成する手法の提案を行う．CAD モデルの持つ性質を利用してモデルの各ポリゴンが投影画像にもたらす画素値の変化を符号付きの深度画像として計算し，これをすべてのポリゴンにおいて累積することで透過投影画像を作成する．モデルの頂点データからボクセルデータへの変換を必要としないため計算量を削減でき，ポリゴンの並び順に依存しないためポリゴンのソート処理が不要で並行処理も可能という利点がある．

これにより，従来と同等の投影作成処理における処理時間とメモリ使用量を抑えることができるため，より解像度の高い投影画像を作成し複雑なモデルの検索に対応が可能となる．さらに特徴量のうち検索結果に及ぼす影響が少ない高周波成分の削減や検索アルゴリ

ズムの改善・並列処理によって、より大規模なモデルの検索を行う。

提案手法を実験的に評価するため、形状の異なる Clutch・Die・Gear の 3 種類のモデルを基に、部品配置の異なるモデルをそれぞれ用意した。評価実験により、提案手法が部品配置の異なるアセンブリモデルを識別可能であることを示す。

本論文は 5 章で構成される。

第 1 章では先述の通り研究の背景および本研究の概要を述べた。

第 2 章では 3 次元モデル検索に関する関連研究について述べる。

第 3 章では本研究の準備として 3 次元アセンブリモデル検索手法の概要について述べる。

第 4 章では本研究で提案する透過投影作成手法について述べる。

第 5 章では透過投影画像の 4 つの作成手法の比較と、提案手法を用いた大規模モデル検索の評価実験について述べる。

第 6 章では本論文のまとめと今後の課題について述べる。

## 2 関連研究

3次元CADモデルを表現するポリゴンメッシュモデルの2次元平面への投影方法についてはこれまでに多く提案されている。レイトレーシング法はStevenら[2]の提案した投影面からの平行光線を仮定して画素値を求める手法である。投影面側に視点を仮定し、視点に及ぶ光線の経路を追跡することで光線の反射や屈折を考慮した写実的な画像を得ることもできる。このうち反射屈折を考慮しない単純な例がレイキャスティング法である。スキャンライン法はWatkins[3]の提案した投影面に垂直なスキャン平面を仮定してスキャンライン上の画素値を求め、スキャン平面を移動させていくことで投影面全体の画素値を求める手法である。3次元CADモデルへの適用においては多田村ら[4]はレイトレーシング法を、床井ら[5]はスキャンライン法を用いて3次元CADモデルの陰影画像を作成している。これらの手法ではモデルの内部構造を反映した透過投影画像を得るためではなく、モデル表面の形状を可視化するために投影画像を作成している点が我々の研究と異なる。

また3次元モデルの検索に関しては、Chenら[7]はモデルの周囲を囲む正20面体の頂点からシルエット画像を作成し“Light Field Descriptor (LFD)”という特徴量から検索を行う手法を提案している。この手法では3次元アセンブリモデルの内部構造を得ることができず、モデルの外観のみに注目している点が我々の研究とは異なる。Huら[8]は3次元アセンブリモデルの構成部品ごとにLFDを求め、部品ごとの類似度で検索を行った。この研究ではモデル間の部品同士のマッチングを行っているが、モデル内の部品配置を考慮した識別までは行っていない点が我々の研究と異なる。

先行研究として片山ら[1]が提案した手法がある。この手法ではモデルの周囲から作成した透過投影画像を元に、モデルの回転と平行移動に対して頑健な特徴量を計算することで、高い精度でアセンブリモデルの検索を行うことができた。しかしモデルのボクセルデータへの変換処理と、解像度の高い投影画像作成時の計算量によって扱うことのできるモデルの規模に制限があった。そのため本稿では投影作成処理と特徴量計算処理において計算量を削減する手法の提案を行う。

### 3 準備

本章では，本稿で行う 3 次元 CAD アセンブリモデルの検索手法の概要について述べる．一般に，3 次元モデルの空間上の姿勢や位置はモデルの製作者によって異なる．そのため，モデル検索にはモデルの回転と平行移動に対して頑健な特徴量が必要となる．また，モデル内部の部品配置を考慮するため，モデルの外見だけでなく内部の情報も含む特徴量が必要である．そこである投影点からモデル空間の中心へ向かう直線に垂直な投影面を仮定し，CT 画像のように内部の構造の違いが画素に表れる透過投影画像を計算する．

#### 3.1 投影画像計算

透過投影画像作成において用いる投影点について，球面座標系における仰角および方位角を等間隔に変化させた点を用いると極付近に投影点が集中するという問題がある．この問題の対策として，投影点には正多面体を拡張して得られる “Geodesic Sphere” の頂点座標を用いる (図 1)．Geodesic Sphere は各頂点間の距離が等しいという特徴を持つため，この頂点を投影点に用いて中心にモデルを配置することで，モデルの周囲から偏りなく投影画像を作成することができる．

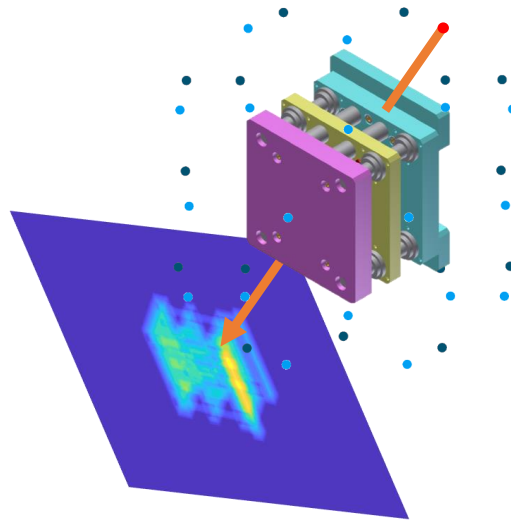


図 1: Geodesic Sphere の頂点からの透過投影画像

### 3.2 特徴量計算

本節では透過投影画像から特徴量を計算する処理について述べる．本稿では同一モデル内で同一の材質ラベルが付与された部品の集合をサブアセンブリと呼ぶ．検索においては部品配置の違い，すなわちサブアセンブリの組み合わせの違いの判別も行うため，投影画像をサブアセンブリごとに個別に作成する (図 2)．

まず 3.1 節の処理により各部品の透過投影画像を作成し，サブアセンブリごとに累積する．この状態ではサブアセンブリ間の位置関係の情報を持たないため，モデル内のどの位置に存在するか情報を付与する．この時，単純に各サブアセンブリの投影を統合すると体積の小さなサブアセンブリの情報が失われるため，注目サブアセンブリの強調された投影画像の生成を行う．ある材質ラベルに着目し，そのラベルが付与されたサブアセンブリと付与されていないサブアセンブリの集合に区別する．それぞれの投影画像に対して体積の逆数を係数として乗算し，2つの投影画像を加算することによって着目サブアセンブリが強調された投影画像を得る．これによって体積の小さなサブアセンブリの情報を失うことなくサブアセンブリ間の位置関係を表現することができる．

こうして得られた投影画像に対してラドン変換を行いサイノグラムを生成する．サイノグラムは原画像の回転を角度方向の平行移動に変換し，平行移動を動径方向の振動に変換する性質がある．

このサイノグラムに対して動径方向に離散フーリエ変換を行い，振幅スペクトルを求める．更に角度方向に離散フーリエ変換を行ったものをサブアセンブリの特徴量とする．この処理によってモデルの回転と平行移動に対して頑健な特徴量を得ることができる．

### 3.3 類似度計算

モデル検索においては，まず入力モデルに対して前述の処理を行い特徴量を計算する．この特徴量とデータベースモデルの特徴量を用いて，サブアセンブリ間の対応付けを行う．各サブアセンブリの特徴量は各投影点から生成された特徴量の集合であるため，更に投影点間の対応付けを行う．これらの対応付けには割り当て問題を解決するアルゴリズムであるハンガリアン法を用いる．本稿では対応付けのコストに各特徴量間のユークリッド距離を用いた．このようにして求めた各特徴量間の距離の総和をモデル間の距離とし，モデル間の距離の逆数をモデル間の類似度としてモデルの検索に用いる (図 3)．

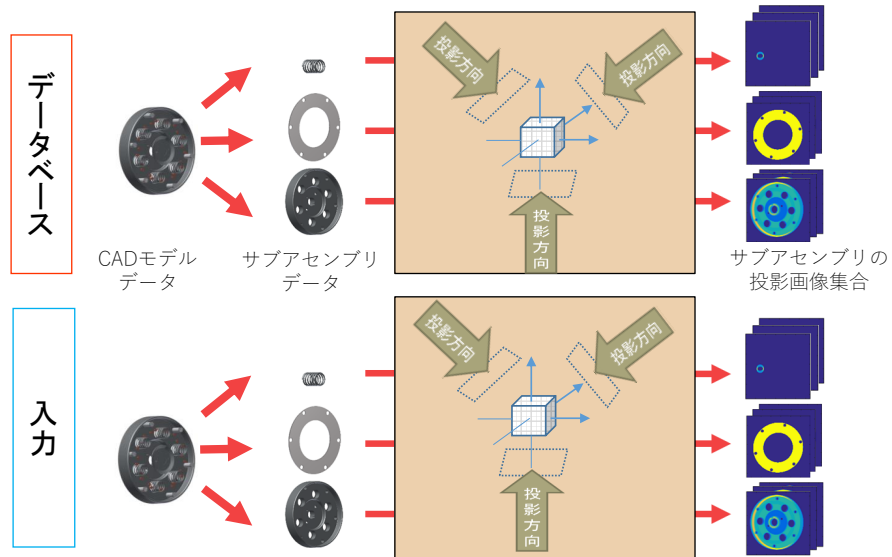


図 2: アセンブリモデルの透過投影画像群作成

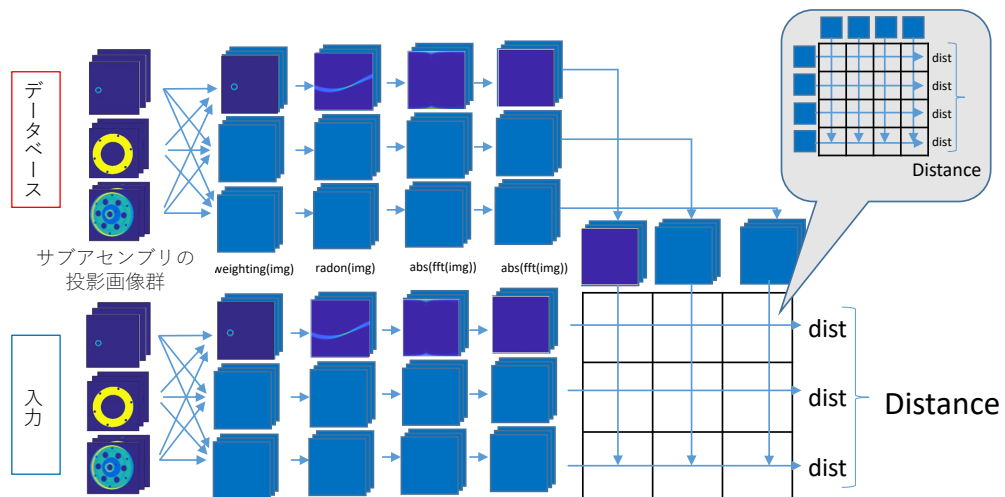


図 3: 透過投影画像からの特徴量計算および類似度計算

## 4 頂点データからの透過投影画像作成処理

### 4.1 はじめに

本章では3次元アセンブリモデルの透過投影画像作成処理について述べる。

3.1 節で述べたように、本研究では3次元モデルから透過投影画像を作成する。先行研究 [1] で用いられた手法は、モデルの頂点データを一度3次元ボクセルデータに変換するものであった。この手法は材質ラベルの違いを3次元配列内の値で表現することができ、直感的に理解しやすいという利点がある。しかしモデルが複雑になると、モデルを表現するための3次元配列のデータ量が配列の1辺の大きさの3乗に比例して増加するという欠点がある。そのため変換処理における処理時間とメモリ使用量を考慮すると、モデル検索において扱えるボクセルの大きさに制限があるという問題があった。

そこで本研究ではボクセルデータへの変換を行わず、頂点データから投影画像を作成し従来手法との処理時間とメモリ使用量の比較を行う。

### 4.2 比較手法

本節では、従来手法と頂点データから投影画像を作成する2つの手法について述べる。

#### 4.2.1 頂点データからボクセルデータへ変換する方法

先行研究で用いられたボクセルデータへの変換を行う投影作成手順は以下のとおりである。Adam [10] の Mesh voxelisation ライブラリを使用した。

1. 頂点データをボクセルデータに変換
2. 回転・平行移動を補間で表現
3. 1方向に要素の和をとり投影を作成

#### 4.2.2 レイキャスティング法

レイキャスティング法による投影作成手順は以下のとおりである。

1. 投影面の画素からの平行光線を仮定する
  2. 光線と各ポリゴンとの衝突判定を行い、交点の座標を求める
  3. 各画素において、記録された交点までの距離を昇順に並び替える
  4. 偶数番目と奇数番目の距離の差を累積することで画素値を求める
  5. 全ての画素において繰り返す
-

### 4.2.3 スキャンライン法

スキャンライン法による投影作成手順は以下のとおりである。

1. XY 平面に平行な平面を仮定する
2. 平面とポリゴンとの交差による線分を記録する
3. 各画素において、記録された線分までの距離を昇順に並び替える
4. 偶数番目と奇数番目の距離の差を累積することで画素値を求める
5. 平面を y 軸方向に移動させ繰り返す

## 4.3 提案手法

本節では、本章で提案するレイキャスティング法を基に考案した透過投影作成手法について述べる。レイキャスティング法及びスキャンライン法では、どちらも各画素の画素値を求める際にモデルと投影面までの距離に応じたソート処理が必要である。ここで投影のある画素における  $2n$  個の交点のうち、昇順ソート後の  $i$  番目の交点の  $z$  座標を  $z(i)$  とすると各画素の画素値  $S$  は次式のように表される。

$$S = \{z(2) - z(1)\} + \{z(4) - z(3)\} + \cdots + \{z(2n) - z(2n-1)\} \quad (1)$$

$$= \sum_{k=1}^n \{z(2k) - z(2k-1)\} \quad (2)$$

$$= \sum_{k=1}^n z(2k) - \sum_{k=1}^n z(2k-1) \quad (3)$$

ソート処理が必要となるのは、全ての交点を求め終わるまで各交点が奇数番目になるのか偶数番目になるのかが定まらないからである。しかし、ここでモデルの持つ性質に注目する。本稿で使用する CAD モデルはソリッドモデルでありポリゴンメッシュで定義されている。また各ポリゴンは3つの頂点で定義され、モデルの内側から外側に向かう法線を持つ。このことから以下の条件が成り立つ。

- ポリゴンがモデル全体で閉じている
  - ポリゴンの間に隙間がない
  - ポリゴンがモデルの内部から外部へ向かう法線を持つ
- ポリゴン同士が交差しない
- シェル数が1である



この条件から、レイキャスティング法のように投影面からの平行光線を仮定すると、各光線がモデルを通過する場合には初めに必ずポリゴンの外面と交差し、またポリゴンの外面と交差した後は必ず別のポリゴンの内面と交差する。つまり

- 光線がポリゴンの外面と交差  $\Rightarrow$  光線がモデルの外部から内部に入る
- 光線がポリゴンの内面と交差  $\Rightarrow$  光線がモデルの内部から外部に出る

この2つが1組の処理として起こることが分かる。さらに式(3)に注目すると、奇数番目の交点がモデル外面との交差、偶数番目の交点が内面との交差であることがいえる。また同式より奇数番目の交点の間、偶数番目の交点の間での順番が画素値に影響しないこともわかる。外面との交点は画素値を投影面との距離分だけ減少させ、内面との交点は画素値を投影面との距離分だけ増加させる働きをする。

この性質を利用してソリッドモデルのポリゴンメッシュの投影画像の作成方法を考案した。イメージを図4に示し、処理についてはAlgorithm1に示す。この手法ではモデルの各ポリゴンが投影画像にもたらす画素値の変化を符号付きの深度画像として計算し、すべてのポリゴンにおいて累積することで透過投影画像を作成する。そのためポリゴンの並びに関わらず同じ投影画像を得ることができる。

---

**Algorithm 1** DepthVolumeFunction(imgSize,vertex)

---

```

1: for all ポリゴン such that ポリゴン  $\in$  モデル do
2:   法線ベクトルを計算
3:   for all y 座標 such that y 座標  $\in$  ポリゴン範囲 do
4:     for all x 座標 such that x 座標  $\in$  ポリゴン範囲 do
5:       if ポリゴンの投影範囲内 then
6:         バッファに z 座標を記録
7:       end if
8:     end for
9:     if 法線ベクトルの z 要素が負 then
10:      バッファの符号を正にして投影画像に累積
11:    else
12:      バッファの符号を負にして投影画像に累積
13:    end if
14:  end for
15: end for

```

---

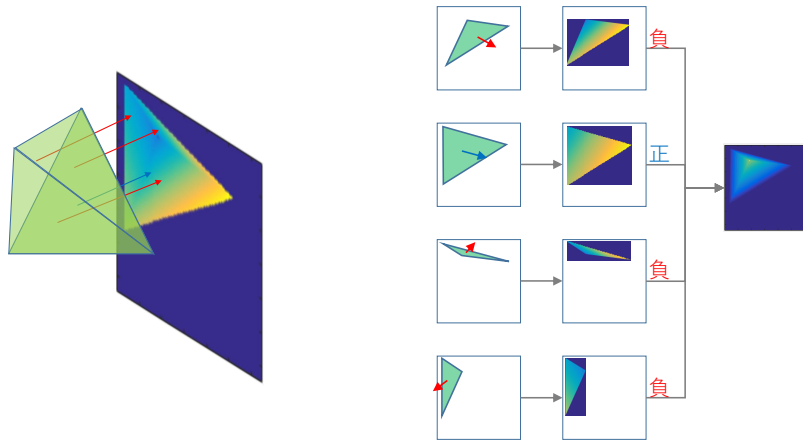


図 4: 提案手法による投影作成

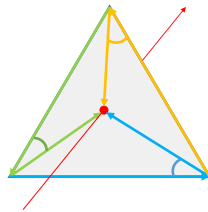


図 5: ポリゴン内外判定

#### 4.3.1 画素値の計算

まず投影面の各画素の中央から投影面に垂直なサンプリング光線を仮定し、判定するポリゴンを含む平面の方程式との交点を符号付きで計算する。この Z 座標の値を投影画像に累積し、これをすべてのポリゴンにおいて行うことでモデル全体の透過投影画像を取得する。

#### 4.3.2 内外判定

1 つのポリゴンの 3 頂点の X 座標の最大値・最小値を  $X_{min}, X_{max}$ 、Y 座標の最大値・最小値を  $Y_{min}, Y_{max}$  とする。各ポリゴンに対して  $X_{min} \leq x \leq X_{max}$  かつ  $Y_{min} \leq y \leq Y_{max}$  を満たす画素において、図 5 のように、ポリゴンの各辺のベクトルと頂点からサンプリング光線に向かうベクトルの外積を計算する。この外積が 3 つとも 0 以上の場合、または 3 つとも 0 以下の場合に、注目ポリゴンと交差するとして平面と直線の方程式から交点の Z 座標を求める。さらに 3 つの外積の符号が同じであればポリゴンの内部を通過、1 つの外積が 0 ならばポリゴンの辺を通過、2 つの外積が 0 ならば頂点を通過として記録する。

#### 4.4 評価実験

本節では、頂点データをボクセルデータへ変換し投影を作成する従来手法と、頂点データから投影を作成する提案手法を含む3つの手法の、合計4つの手法を用いて3次元アセンブリモデルの透過投影画像を作成し、処理時間とメモリ使用量の比較を行う。

##### 4.4.1 使用するモデル

実験に使用するモデルは複数のサブアセンブリから構成されたアセンブリモデルであり、サブアセンブリはさらに複数の小さな部品モデルの集合によって構成されている。各部品モデルは3次元空間上の3つの頂点からなるポリゴンの集合で表現されたポリゴンメッシュモデルである。使用モデルはGrabCAD [9] から取得した Clutch (55 部品), Die (39 部品), Gear (55 部品) の3種類である(図6)。投影枚数は Geodesic Dome の頂点の162点、投影の1辺の画素数は32, 64, 128, 256[px]の4種類である。

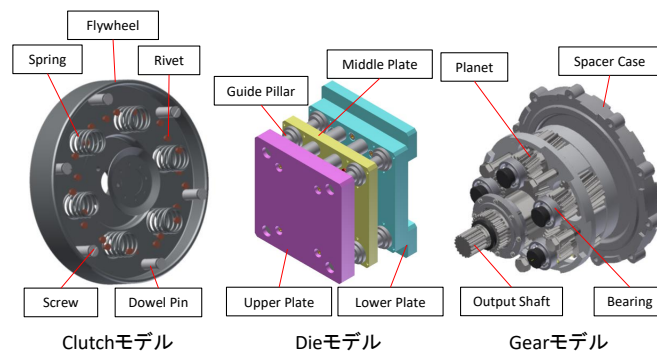
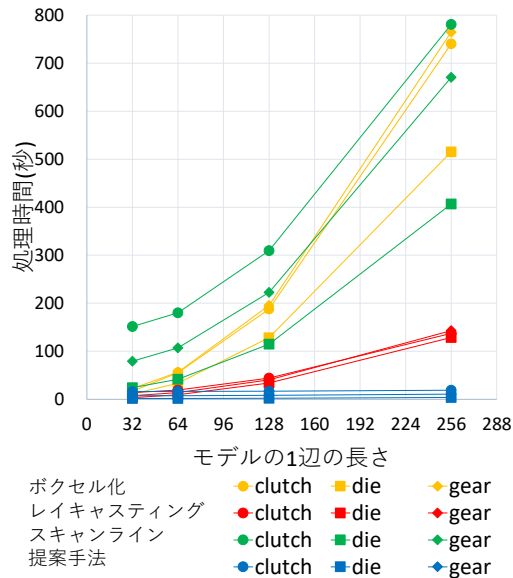


図 6: 実験に使用するモデルと特徴的な部品

##### 4.4.2 処理時間の比較

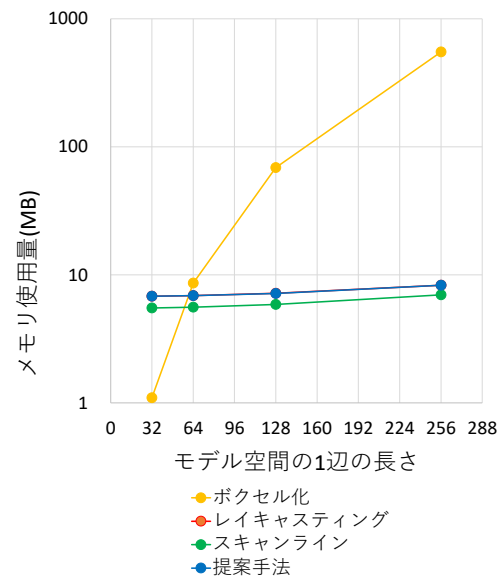
それぞれのモデルについて投影手法と投影の大きさを変化させ処理時間を計測した。結果を図7に示す。

各モデル間の処理時間を比較すると、どの手法と投影の大きさにおいても Die モデルが処理時間が短く、次いで Clutch モデル、Gear モデルの順に処理時間が長くなる傾向がある。モデルの形状によって特定の手法が有効である結果は見られなかった。同一モデルで各手法間の処理時間を比較した場合、投影の大きさが小さい場合はボクセル化経由と深度バッファを利用した方法が同程度に短いですが、投影の大きさが大きい場合にはスキャンライ



各投影手法における処理時間

図 7: 処理時間の比較



各投影手法におけるメモリ使用量

図 8: メモリ使用量の比較

ン法と深度バッファ法が有効になる。ボクセル化経由とレイキャスティング法においては投影の大きさを大きくしたときの処理時間の増え方が著しく、投影の1辺の長さが2倍になると処理時間は4倍程度になる。一方でスキャンライン法と深度バッファ法では投影の1辺の長さに対する処理時間の増加は線形的である。この結果から、256×256よりも大きな投影を作成する場合にもスキャンライン法と深度バッファ法が有効であると考えられる。

#### 4.4.3 メモリ使用量の比較

次に、Clutch モデル, 投影サイズ 128px×128px での各手法におけるメモリ使用量を計測した。その結果を図8に示す。

ボクセル化経由ではモデルの回転を表現するためにモデルを一度 (投影の大きさ×投影の大きさ×投影の大きさ) の3次元配列に格納し、さらに回転後のボクセルデータを別に用意した配列にもう一度格納する必要があるため非常に大きなメモリを必要とする。一方で頂点データから投影を作成する手法ではモデルの回転を頂点の座標変換で表現することができるため、投影を作成するためのメモリ使用量を削減できるという特徴がある。

## 5 投影作成処理の改善

### 5.1 はじめに

本項では透過投影作成処理における問題の解決，およびアルゴリズム改良による処理の高速化について述べる．また従来手法および改善前後の処理時間について比較実験を行い，性能を評価する．

### 5.2 提案手法

#### 5.2.1 微小ポリゴンの処理

画素の中心間の距離に対して十分に微小なポリゴンの場合，どのサンプリング光線とも交差せず，投影画像の画素値に変化をもたらさない場合がある．図9において方眼の中心をサンプリング点としたとき，最も左側の2つのポリゴン以外は投影画像に変化をもたらさない．ポリゴンの3頂点のX座標の最大値・最小値を  $X_{min}, X_{max}$ ，Y座標の最大値・最小値を  $Y_{min}, Y_{max}$  とすると， $\text{ceil}(X_{min}) \leq \text{floor}(X_{max})$  または  $\text{ceil}(Y_{min}) \leq \text{floor}(Y_{max})$  を満たす場合は必ずサンプリング点を通らない．これを利用して枝刈りを行うことによって計算量を削減することができる．図9においては最も右側の3つのポリゴンは以降の処理を行わない．

#### 5.2.2 ポリゴン境界の処理

図10のように，求めた画素値に係数として，光線がポリゴンの内側を通る場合は1，辺を通る場合は0.5，頂点を通る場合は画素のうち占める面積を乗算する．この処理によって光線とモデルの交差1回あたりのサンプリング量にあたる係数の総和が1になり，ポリゴン境界のサンプリングの重複やポリゴンの処理順による画像値の変化を防ぐことが可能である．また光線がモデルの表面を掠める場合でも同様の処理で解決できる．

#### 5.2.3 同次変換行列の利用

従来手法においてはモデルの回転・拡大縮小・平行移動，および投影点の移動をそれぞれ座標変換と補間によって計算していた．提案手法では同次変換行列を用いて変換行列をまとめ座標変換の回数を減らすことで，計算誤差を小さくするとともに計算量も削減した．

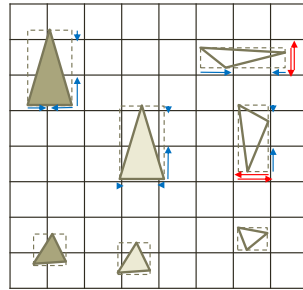


図 9: 微小ポリゴンの処理

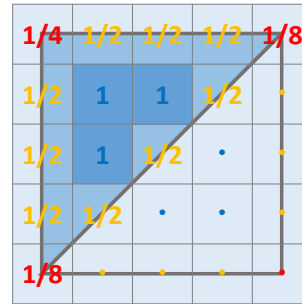


図 10: ポリゴン境界の処理

### 5.3 評価実験

本節では、改善した提案手法において、従来手法および改善前の提案手法との比較を行う。それぞれの手法を用いて複数の解像度の透過投影画像を作成し、処理時間を比較し評価する。

#### 5.3.1 使用するモデル

実験に使用するモデルは複数のサブアセンブリから構成されたアセンブリモデルであり、サブアセンブリはさらに複数の小さな部品モデルの集合によって構成されている。各部品モデルは3次元空間上の3つの頂点からなるポリゴンの集合で表現されたポリゴンメッシュモデルである。使用モデルはGrabCAD [9] から取得したClutch (55 部品), Die (39 部品), Gear (55 部品) の3種類である(図 11)。従来手法および改善前の提案手法では、投影点数はGeodesic Domeの頂点の12点、投影の1辺の画素数は32, 64, 128, 256[px]の4種類である。改善した提案手法において、投影点数はGeodesic Domeの頂点の12, 42, 92, 162, 252点、投影の1辺の画素数は32, 64, 128, 256, 512[px]の4種類である。

#### 5.3.2 処理時間の比較

それぞれのモデルについて投影手法と投影の大きさを変化させ処理時間を計測した。結果を図 12 に示す。従来手法と比較すると、提案手法は処理時間の増加が緩やかである事が分かる。さらに改善後の提案手法ではより難しい条件での投影作成も行った。その結果、投影の1辺の画素数を512px、投影点数を252点まで増加させても、従来手法の投影の1辺の画素数128px、投影点数を12点よりも高速である。また提案手法の改善前後の比較では、Clutchモデルでは4倍程度、Die, Gearモデルでは2倍程度の高速化が実現できた。

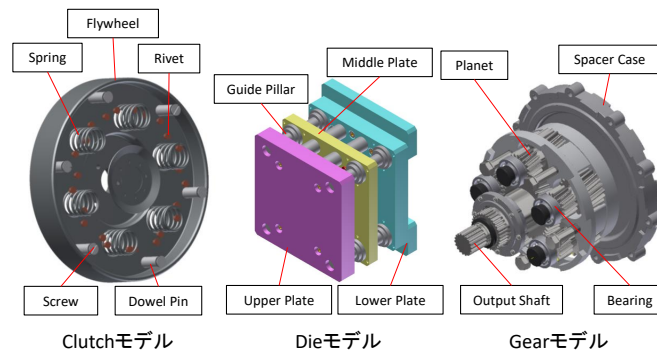


図 11: 実験に使用するモデルと特徴的な部品

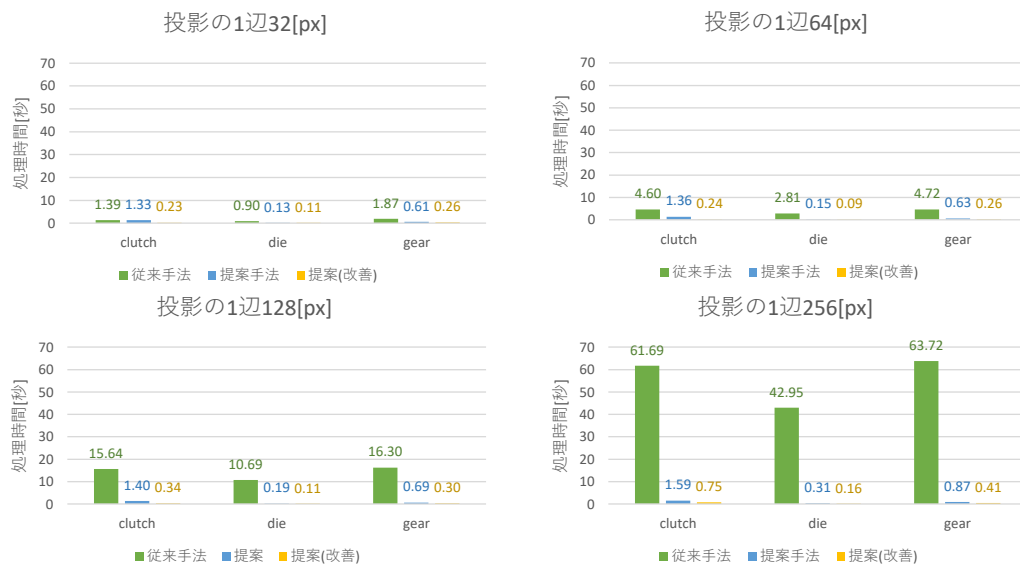


図 12: 各モデルの投影作成処理時間

## 6 大規模なモデルの検索

### 6.1 はじめに

本章ではサブアセンブリ数の多い大規模なモデル検索への対応について述べる．多数の部品の組み合わせからなる大規模なモデルの検索では，小さな部品が投影画像に反映されるように解像度を上げる必要がある．5章までに述べた投影作成手法を用いて大規模なモデルの検索を行い性能を評価する．

### 6.2 大規模なアセンブリモデルの特徴量計算

先行研究においては強調された投影画像作成のために，あらかじめ全てのサブアセンブリの投影画像を作成した後に重み付けを行っていた．しかし大規模なモデルの検索においては，投影画像の解像度の増大やサブアセンブリ数が増加に伴って，多数の投影画像をメモリ上に保持しておくことは難しくなる．例として投影画像の1辺の長さが4096px，投影枚数92枚，サブアセンブリ数40の条件でモデル1個の投影画像を保持するには  $4096px \times 4096px \times 8 \text{ バイト} \times 92 \text{ 枚} \times 40 \text{ サブアセンブリ} = 460 \text{ ギガバイト}$  の記憶領域が必要となる．

そこで本稿ではフーリエ変換の性質を利用して特徴量の削減を行う．離散フーリエ変換は周期性を持ち，振幅スペクトルをとると原点を挟んで対称の関係になる．この繰り返し部分を考慮することで特徴量を1/2に削減する．

また離散フーリエ変換は線形性を持つため， $p_1(n) \sim p_m(n)$  を元画像， $a_1 \sim a_m$  を重みとして次のように表される

$$\mathcal{F}[aP(n)] = aP(k) \quad (4)$$

$$\mathcal{F}\left[\sum_{i=1}^m a_i P_i(n)\right] = \sum_{i=1}^m (a_i P_i(k)) \quad (5)$$

これによりあらかじめサイノグラムに対して離散フーリエ変換を行い，サブアセンブリの体積の逆数を重みとして乗算したものを累積していくことでも従来と同様の特徴量が求められる事が分かる．

また本手法においてサイノグラムのフーリエ変換後の特徴量では低周波成分にほとんどの値が集中し，高周波成分は類似度計算にてほぼ影響を与えないという特徴がある．そのため高周波成分を削減することで類似度計算にほぼ影響がないままメモリ使用量を削減することができる．さらに，前述の周期性を利用した特徴量削減と合わせて，以降の計算量



の削減にもつながる。本稿では動径方向において第 128 項まで、角度方向においては第 30 項までの係数を用いた。これによってモデル 1 個あたりに必要な記憶領域は  $128\text{px} \times 30\text{px} \times 8\text{バイト} \times 92\text{枚} \times 40\text{サブアセンブリ} \approx 0.1\text{ギガバイト}$  まで削減できる。

### 6.3 評価実験

実験モデルとして複数のモデルを組み合わせてサブアセンブリ数の多い複雑なモデルを作成し、透過投影画像の作成、特徴量計算、類似度計算を行う。また、処理時間や正解率、投影の精度についても評価を行う。

#### 6.3.1 使用するモデル

実験に使用するモデルは複数の小さなモデルを組み合わせてサブアセンブリ数を増やしたアセンブリモデルである。サブアセンブリはさらに複数の小さな部品モデルの集合によって構成されている。各部品モデルは 3 次元空間上の 3 つの頂点からなるポリゴンの集合で表現されたポリゴンメッシュモデルである。使用モデルは GrabCAD [9] から取得した Clutch (55 部品), Die (39 部品), Gear (55 部品) の 3 種類をもとにさらに部品配置の異なるモデルを 5 種類ずつ、計 15 種類のモデルを組み合わせて作成する。(図 13, 図 14 図 15)。同一モデル内で同じ色を持つ部品は共通のサブアセンブリに属することを示す。

データベースモデルの投影点数は Geodesic Dome の頂点の 92 点、クエリモデルはの投影点数は Geodesic Dome の頂点の 12 点である。

部品数が多い実験モデルとして、前述の部品配置の異なるモデルをモデル空間内に並べ、多数のサブアセンブリからなる大規模な実験モデルを用意した。組み合わせのイメージを図 16 に示す。大規模な実験モデルとして、モデル空間内に Clutch3 個, Die3 個, Gear2 個の合計 8 個のモデルを並べたものを 1 つの大規模なクエリモデル 001 とする。Clutch, Die, Gear はそれぞれ 5 種類のサブアセンブリを持つため、このモデルは 40 個のサブアセンブリから構成される。これを基準として、クエリ 002 は全体の形状が同じだが一部のサブアセンブリの配置が異なるモデル、クエリ 003 は 8 個のモデルの位置が入れ替わっているもの、クエリ 004 はすべてのサブアセンブリの大きさを縮小したもの、クエリ 005 は Clutch モデルと Die モデルを 1 つずつ取り除いたもの以上の計 5 種類を実験用の大規模なモデルとして用意した。また、モデル空間内の 1 辺に 3 個、合計 27 個のモデルを配置したサブアセンブリ数 135 個のモデル、モデル空間内の 1 辺に 4 個、合計 64 個のモデルを

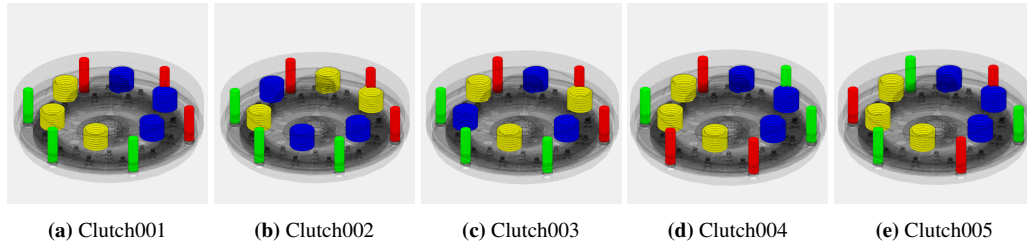


図 13: Clutch モデル

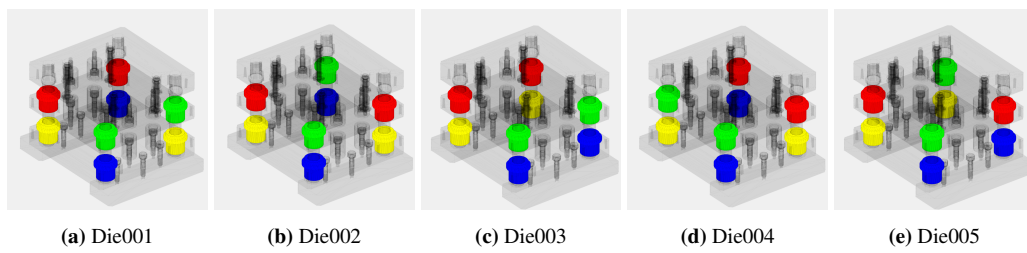


図 14: Die モデル

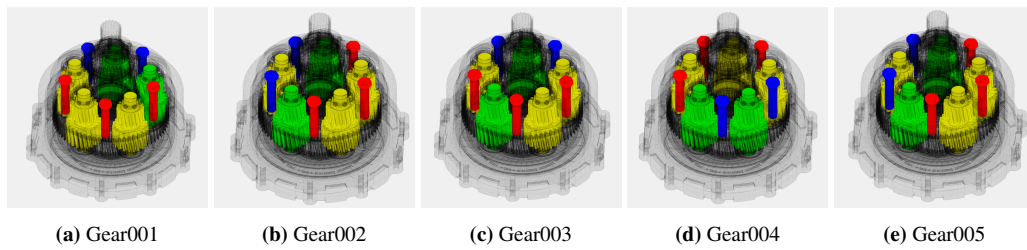


図 15: Gear モデル

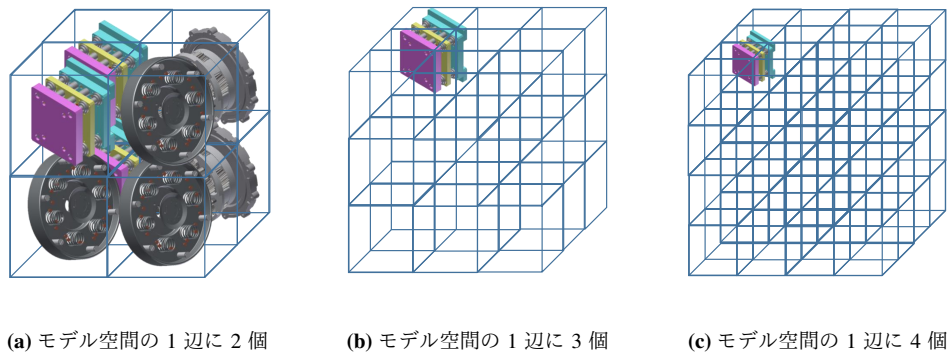


図 16: 大規模な実験モデル

配置したサブアセンブリ数 320 個の大規模なモデルについても同様に作成しそれぞれ検索を行う。

### 6.3.2 処理時間の比較

各段階の処理時間について、投影サイズとサブアセンブリ数によるモデル 1 個あたりの処理時間の変化を次の図 17, 図 18, 図 19 に示す。投影画像作成処理およびサイノグラム作成処理においては、投影画素数が処理時間に大きく影響を与える事が分かる。またフーリエ変換の性質を利用した特徴量削減により、特徴量計算と類似度計算では投影の解像度が高くても処理時間を抑えることができた。

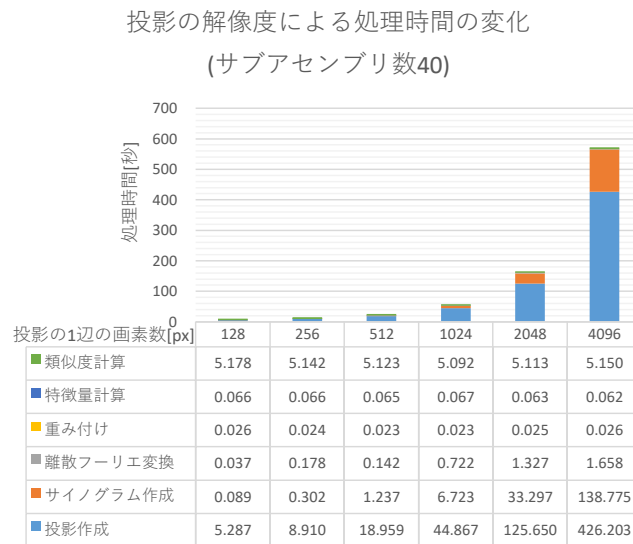


図 17: 大規模な実験モデル検索の処理時間 (サブアセンブリ数 40)

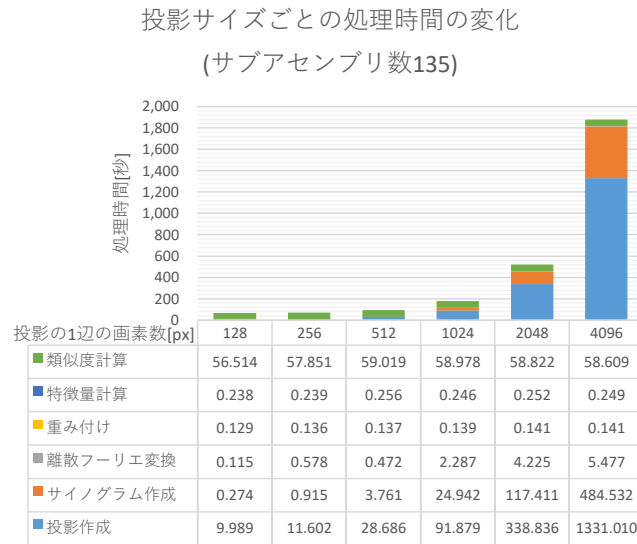


図 18: 大規模な実験モデル検索の処理時間 (サブアセンブリ数 135)

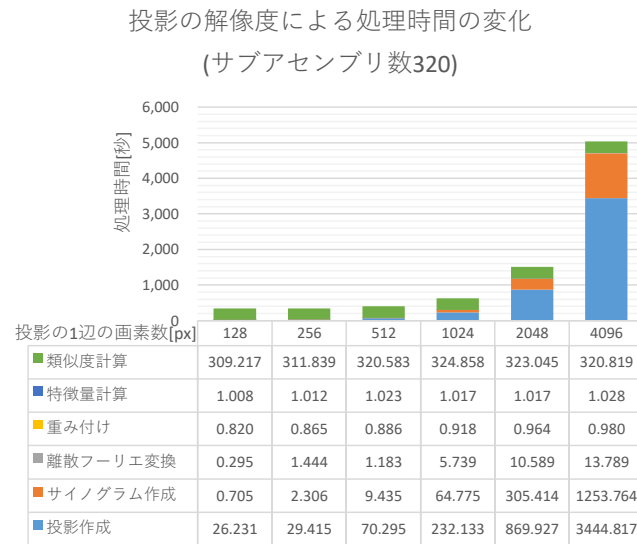


図 19: 大規模な実験モデル検索の処理時間 (サブアセンブリ数 320)

### 6.3.3 正答率

部品配置の異なる5種類のクエリモデルを入力したとき、データベース内にあるサブアセンブリの配置が同じモデルを正解モデルとして、クエリモデルと正解モデルとの類似度が他のどのデータベースモデルよりも高い場合を「正答」としたときの正答率を次の図 20 に示す。この図ではモデルの複雑さに対して投影解像度が高い、サブアセンブリ数 40 かつ投影の 1 辺の画素数 2048 および 4096 ではどちらも正解率 100%と高くなっていることが読み取れる。しかし、モデルの複雑さに対する投影解像度が最も低いサブアセンブリ数 320 かつ投影の 1 辺の画素数 128 においても 80%という正解率が出ている。

誤答の内訳の分析のため各モデルの検索における類似度に着目する。最も正解率が低かった 1 辺の画素数 128, サブアセンブリ数 135 の類似度を図 21, 最も正解率が高かった 1 辺の画素数 4096, サブアセンブリ数 40 の類似度を図 22 に示す。

この図から形状の似ているクエリ 001, 002, 003 に対して, 004 とのおよび 005 との違いについてはどちらの条件でも正しく判別できている事が分かる。これはここに示した図以外においても同様で, 大きく形状の異なるモデルについては正しく検索ができるといえる。それに対してモデルの形状が似ていて細かい部品配置のみが異なるクエリ 001, 002, 003 の間では特徴量が近くなりやすく, 特にクエリ 001 と 003 の対応付けを間違えるケースが多かった。

### 6.3.4 誤差

6.3.3 において誤答であった場合に, どの程度間違っていたのか検討する。入力モデルと正解のデータベースモデルとの距離と, 入力モデルと類似度計算によって選んだモデルとの距離との差を計算し, その差を奥行方向の距離である投影の 1 辺の画素数で割ったものを誤差とする。5 種類の入力モデルの誤差の合計を求め, 図 23 に示す。この図から, 投影解像度が高い図の右側ほど, モデル空間内の奥行に対して類似度の誤りの割合が低い事が分かる。また, モデル内の部品数が多く複雑なモデルであるサブアセンブリ数 320 のグラフは投影解像度が低くなるほど誤りの割合が高くなっていくことが読み取れる。

以上より, 投影の解像度は詳細な投影画像の作成が可能になるとともに, 後の特徴量計算, 類似度計算においても検索精度の向上に貢献しているといえる。また, モデルの複雑さに対して投影解像度が一定を下回ると誤差が大きくなることもいえる。

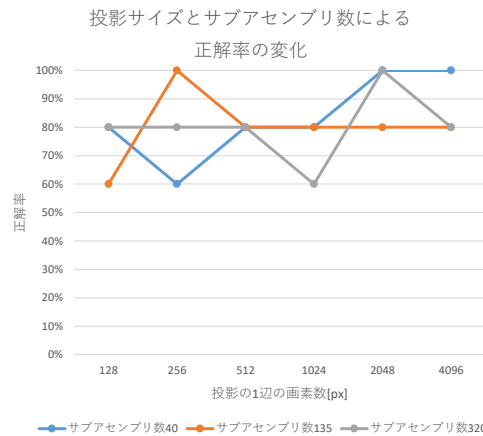


図 20: モデルの規模と投影解像度による検索正解率の変化

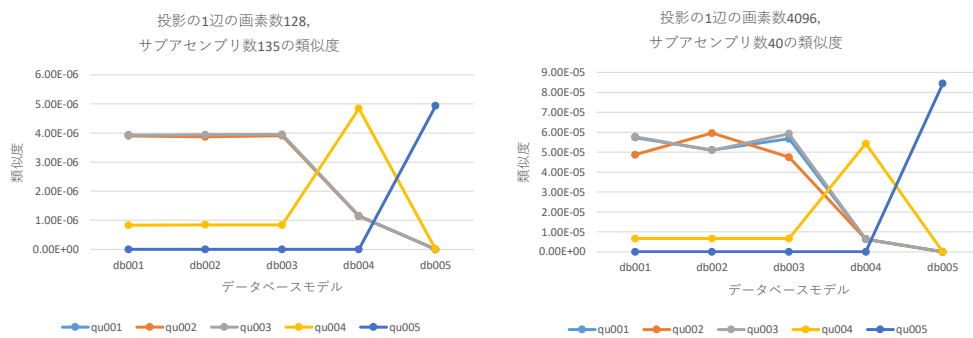
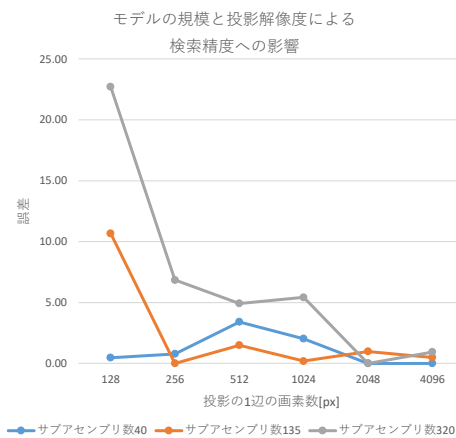
図 21: 1 辺の画素数 128, サブアセンブリ数 135 の類似度  
図 22: 1 辺の画素数 4096, サブアセンブリ数 40 の類似度

図 23: モデルの規模と投影解像度による検索精度への影響

## 6.4 同一形状を持つ大規模なモデルの検索

6.3.3 項で使用したモデルはそれぞれ全体として異なる形状であったが、本節では全体として同じ形状を持つモデルの検索を行う。使用するモデルはモデル空間内のモデル数が8個、27個、64個、125個の4種類。それぞれに形状が同じで部品配置の異なるモデル10種を用意した。投影枚数はデータベース側が92枚、クエリ側が12枚、投影画像の1辺の画素数は256, 512, 768[px]の3種類で行った。10種類のモデルの入力に対するデータベースモデル10個との類似度計算を行い、結果を類似度の高い順に5つ並べたものを図24に示す。表内の正解モデルの位置を強調して示し、10種類の入力に対する正解モデルの平均順位と、正解モデルの類似度が最も高くなった割合も併せて示す。

同一解像度内のサブアセンブリ数の違いに注目すると、モデル空間内のサブアセンブリ数が少ないほど正解モデルの平均順位が高くなることがわかる。またサブアセンブリ数が多い場合においては投影画像の解像度が低いほど正解モデルの平均順位が低くなっている。この結果から、モデルの規模に対して投影画像の解像度が不十分である場合には検索精度が低下することがあり、大規模なモデルの検索において更に複雑なモデルを扱うためにはそれに応じた更なる投影の解像度が必要であると推測される。

256px×256px														512px×512px														768px×768px													
8個	入力	1	2	3	4	5	6	7	8	9	10	1.3	入力	1	2	3	4	5	6	7	8	9	10	1.4	入力	1	2	3	4	5	6	7	8	9	10	1.6					
	1st	1	2	9	4	5	6	7	8	9	2	80%	1st	3	10	3	4	5	6	7	8	3	10	70%	1st	9	10	3	4	5	6	7	1	1	2	50%					
	2nd	9	10	1	9	3	1	3	1	1	10		2nd	1	2	9	3	3	9	3	1	1	2		2nd	1	2	1	1	1	3	1	3	9	10						
	出力3rd	3	3	3	1	1	9	1	3	3	1		出力3rd	9	3	1	9	7	1	9	3	9	3		出力3rd	3	1	9	3	9	9	9	4	3	3						
	4th	7	1	7	3	7	3	9	7	4	3		4th	7	1	7	1	9	3	1	9	8	9		4th	4	3	4	7	3	1	3	8	4	1						
	5th	10	9	2	7	9	4	8	9	8	4		5th	8	9	4	8	1	8	8	7	4	1		5th	7	9	7	9	7	4	4	9	7	7						
27個	入力	1	2	3	4	5	6	7	8	9	10	1.9	入力	1	2	3	4	5	6	7	8	9	10	1.8	入力	1	2	3	4	5	6	7	8	9	10	3.1					
	1st	9	9	10	4	5	6	9	9	9	10	50%	1st	9	2	10	1	5	6	1	8	1	10	50%	1st	6	7	10	7	5	6	7	7	8	10	50%					
	2nd	2	1	3	1	4	9	4	8	1	3		2nd	2	9	3	4	1	9	9	9	2		2nd	8	9	1	9	1	8	6	8	1	2							
	出力3rd	1	2	4	9	1	1	1	6	4	2		出力3rd	4	1	9	9	4	1	7	2	8	3		出力3rd	7	8	3	1	9	7	9	9	6	3						
	4th	8	10	1	8	7	4	7	1	6	1		4th	1	8	5	8	8	4	2	4	4	8		4th	2	2	7	8	8	9	4	1	9	6						
	5th	4	4	9	7	9	8	8	4	8	9		5th	6	4	8	2	2	8	4	1	2	4		5th	9	10	9	6	2	1	8	2	7	8						
64個	入力	1	2	3	4	5	6	7	8	9	10	3.4	入力	1	2	3	4	5	6	7	8	9	10	1.4	入力	1	2	3	4	5	6	7	8	9	10	2.7					
	1st	9	4	7	4	5	6	9	9	9	10	50%	1st	7	10	9	4	5	3	3	1	3	10	30%	1st	10	2	9	9	1	5	7	1	8	2	10%					
	2nd	10	9	9	9	9	9	7	1	3	6		2nd	4	2	7	9	9	6	1	4	1	2		2nd	3	1	3	7	5	6	3	9	7	10						
	出力3rd	5	10	1	7	7	10	4	7	7	9		出力3rd	3	7	3	3	3	9	7	8	8	9		出力3rd	7	3	1	1	7	1	9	8	1	9						
	4th	6	7	6	6	3	2	10	10	10	7		4th	1	3	1	7	1	1	10	7	9	7		4th	1	10	10	3	4	10	4	10	5	3						
	5th	4	6	4	3	10	4	6	6	6	2		5th	2	1	8	10	4	7	9	2	4	8		5th	9	7	7	4	2	8	8	7	9	1						
125個	入力	1	2	3	4	5	6	7	8	9	10	4.1	入力	1	2	3	4	5	6	7	8	9	10	3.5	入力	1	2	3	4	5	6	7	8	9	10	3.0					
	1st	9	5	6	5	5	2	7	5	1	9	20%	1st	9	2	10	7	5	9	2	5	2	7	20%	1st	3	10	3	4	8	6	6	8	3	2	40%					
	2nd	7	2	9	9	2	9	9	8	3	7		2nd	2	5	3	4	2	2	9	2	10	2		2nd	7	2	9	1	3	8	3	9	2	10						
	出力3rd	1	1	1	6	8	7	6	2	7	3		出力3rd	10	1	6	1	9	3	1	8	1	9		出力3rd	4	3	8	3	10	2	7	7	9	9						
	4th	3	8	7	3	3	3	3	6	6	1		4th	7	8	1	6	8	10	7	1	9	1		4th	8	8	2	10	5	1	9	3	8	3						
	5th	2	3	5	10	9	1	1	1	9	6		5th	1	7	9	10	6	6	10	7	7	3		5th	2	1	6	8	9	3	1	4	10	8						

図 24: 同一形状を持つ大規模なモデルの検索

## 7 結論

本稿では、透過投影画像から計算した特徴量を用いた 3 次元 CAD アセンブリモデルの部品配置を考慮した部分検索手法を提案した。投影作成処理においては、レイキャスティング法をもとに深度画像を用いた頂点データからの透過投影作成手法を提案した。また、不要処理の削減、同次変換行列による座標変換、ポリゴン境界処理の改善により、投影作成の処理時間とメモリ使用量を削減した。

特徴量計算処理においては、フーリエ変換の性質を利用した処理順の最適化や特徴量の削減によりメモリ使用量と処理時間の削減を行った。

これらの貢献によって従来手法では対応が不可能だった解像度の高い投影を作成することが可能になり、多数のサブアセンブリによって構成される大規模なアセンブリモデルの検索を行うことができた。

今後の課題としてはモデル空間内のサブアセンブリ数を増やしたより大規模なモデルや、データベース内のモデルの数を増した場合など、より難しい条件での検索への対応を考えている。また実用化に向けて GPU の利用、並列処理化による更なる高速化や検索精度の向上も必要となる。



## 8 謝辞

本修士論文は、筆者が首都大学東京大学院システムデザイン研究科情報通信システム学域博士前期課程在学中に片山研究室において行った研究をまとめたものである。本研究に関しお世話になった方々にこの場を借りて感謝の意を表する。

本研究の遂行にあたり、終始ご指導ご鞭撻頂いた片山薫准教授に心より感謝申し上げる。また副査として本論文をご精読いただき、本論文の改善にご助言を頂いた石川博教授、小町守准教授に感謝申し上げる。

マイナ エルネスト ウェケ氏には片山研究室での生活とコーディング技術の両面において多大なご助力を頂いた。深く感謝の意を表する。また、同じ研究室に在籍し共に有意義な時間を過ごした片山研究室の皆様に、深く感謝する。

最後に、私に修士課程に進学する機会を与え、離れた土地から温かく見守り続けてくれた両親に心より感謝する。

## 参考文献

- [1] K.Katayama and T.Sato, “Matching 3D CAD Assembly Models with Different Layouts of Components using Projections”, IEICE Transactions on Information and Systems, Vol. E98–D(6), pp. 1247–1250, 2015.
  - [2] Steven M. Rubin, and Turner Whitted’. “A 3-dimensional representation for fast rendering of complex scenes”. ACM SIGGRAPH Computer Graphics. Vol. 14. No. 3. ACM, 1980.
  - [3] Gary S. Watkins. “A real time visible surface algorithm”. No. UTEC-CSC-70-101. UTAH UNIV SALT LAKE CITY SCHOOL OF COMPUTING, 1970.
  - [4] 多田村克己, 中前栄八郎 and 西田友是. “局所レイトレーシングによる陰影表示法”. 情報処理学会論文誌, Vol. 27, No. 11, pp. 1077–1085, 11 1986.
  - [5] 床井浩平 and 北橋忠宏. “凸な立体の集合演算によって定義された形状のスキャンライン法による陰影画像生成”. 情報処理学会論文誌, Vol. 30, No. 1, pp. 81–90, 1 1989.
  - [6] 川端敦, 渡辺範人 and 坂井俊雄, “マルチ・パス・レンダリング法を用いた高速画像生成手法”, 情報処理学会研究報告グラフィクスと CAD, No. 62, pp. 39–46, 7, 1992.
  - [7] D.Y. Chen, X.P. Tian and Y.T. Shen, M. Ouhyong, “On Visual Similarity Based 3D Model Retrieval”, Computer Graphics Forum, Vol. 22, No. 3, pp. 223–232, 2003.
  - [8] K.M. Hu, B. Wang and J.H. Yong, J.C. Paul, “Relaxed lightweight assembly retrieval using vector space model”, Computer-Aided Design, Vol. 45, No. 3, pp. 739–750, 2013.
  - [9] GrabCAD, <http://grabcad.com/library/>
  - [10] Adam H. Aitkenhead, “Mesh voxelisation”, <https://jp.mathworks.com/matlabcentral/fileexchange/27390-mesh-voxelisation>, 4, 2010.
-